

Complexity Jumps In Multiagent Justification Logic Under Interacting Justifications

Antonis Achilleos

The Graduate Center of The City University of New York,
365 Fifth Avenue New York, NY 10016 USA
aachilleos@gc.cuny.edu

Abstract. The Logic of Proofs, LP, and its successor, Justification Logic, is a refinement of the modal logic approach to epistemology in which proofs/justifications are taken into account. In 2000 Kuznets showed that satisfiability for LP is in the second level of the polynomial hierarchy, a result which has been successfully repeated for all other one-agent justification logics whose complexity is known.

We introduce a family of multi-agent justification logics with interactions between the agents' justifications, by extending and generalizing the two-agent versions of the Logic of Proofs introduced by Yavorskaya in 2008. Known concepts and tools from the single-agent justification setting are adjusted for this multiple agent case. We present tableau rules and some preliminary complexity results. In several cases the satisfiability problem for these logics remains in the second level of the polynomial hierarchy, while for others it is PSPACE or EXP-hard. Furthermore, this problem becomes PSPACE-hard even for certain two-agent logics, while there are EXP-hard logics of three agents.

1 Introduction

Justification Logic is a family of logics of justified beliefs. Where epistemic modal logic treats formulas of the form $K\phi$ with the intended meaning that an agent knows/believes ϕ , in Justification Logic we consider formulas of the form $t:\phi$ with the intended meaning that t is a justification for ϕ - or that the agent has justification t for ϕ . The first Justification Logic was LP, the logic of proofs, and appeared in [4] by Artemov, but it has since developed in a wide system of explicit epistemic logics with notable complexity properties that significantly differ from the corresponding modal logics: while every single-agent justification logic whose complexity has been studied has its derivability problem in Π_2^P (the second level of the polynomial hierarchy), the corresponding modal logics have PSPACE-complete derivability problems. Furthermore certain significant fragments of these justification logics have an even lower complexity- NP, or even P in some cases. For an overview of Justification Logic see [5]. For

an overview of complexity results of (single-agent) Justification Logic, see [10].

In epistemic situations we often have multiple agents, so as it is the case with modal logic, there is a need for a multi-agent justification logic. In [15], Yavorskaya presents two-agent variations of LP. These logics feature interactions between the two agents' justifications: for LP_\uparrow , for instance, every justification for agent 1 can be converted to a justification for agent 2 for the same fact and we have the axiom $t_1\phi \rightarrow \uparrow t_2\phi$,¹ while $\text{LP}_!$ comes with the extra axiom $t_1\phi \rightarrow !t_2t_1$, so agent 2 is aware of agent 1's justifications.

In [3], we extended Yavorskaya's logics to two-agent variations of other justification logics, as well as to combinations of two different justification logics. We then gave tableau procedures to prove that most of these logics were in the second step of the polynomial hierarchy, an expected result which mimics the ones for single-agent justification logics from [9, 10, 1]. For some cases, however, we were able to prove PSPACE-completeness, which was a new phenomenon for Justification Logic.

In this paper we continue our work from [3]. We provide a general family of multi-agent logics. Each member of this family we call $(J_{D,F,V,C}^n)_{CS}$, where n, D, F, V, C are parameters of the logic. For $(J_{D,F,V,C}^n)_{CS}$ we consider n agents and the interactions between the agents' justifications are described by binary relations on agents, V and C . Furthermore, not all agents are equally reliable: D and F are sets of agents, all agents in D have consistent beliefs and all agents in F have true beliefs. These concepts are made precise in section 2. It is our goal to provide a flexible system capable of modelling situations of many diverse agents, or diverse types of justifications, allowing for reasonably general interactions among their justifications.

For this family of logics we provide semantics and a general tableau procedure and then we make observations on the complexity of the derivation problem for its members. In particular, we demonstrate that all logics in this family have their satisfiability problem in NEXP - under reasonable assumptions. This family demonstrates significant variety, as it also includes PSPACE- and EXP-complete members, while of course some of its members have their satisfiability problem in Σ_2^P . This is a somewhat surprising result, as all single-agent justification logics whose complexity is known have their satisfiability problem in Σ_2^P .

This paper is organised as follows. In section 2, we give the base definitions of the syntax, axioms and semantics for each logic in the family.

¹ We take some liberty with the notation to keep it in line with this paper.

Then we reintroduce the star calculus, an invaluable tool, and our first complexity results, mirroring the ones for single-agent justification logic (see [8]). The version of the star calculus we provide is somewhat different than the usual ones in that it is based on a given frame. If the frame includes a single world, we get the usual, more familiar version. In section 3 we give general tableau rules for each of our logics. Naturally, the rules are parameterized by the logic's parameters, including the interactions between the agents, so special attention is given in that section to these interactions. We then go on and further optimize the tableau procedure with respect to the number of world-prefixes it produces; this results in a Σ_2^P upper bound for the satisfiability of a general class of logics.

2 Multiagent Justification Logic with Interactions

In this section we present the system we study in this paper, its semantics and the basic tools we will need later on. Most of the proofs for the claims here can be adjusted from the one- or two- agent versions of Justification Logic. The reader can see [5] or [10] for an overview of single-agent justification logic and [3] for a two-agent version of this system.

2.1 Syntax

In this paper, if $n \in \mathbb{N}$, $[n]$ will be the set $\{1, 2, \dots, n\}$. For every $n \in \mathbb{N}$, the justification terms of the language L_n will include all constants c_1, c_2, c_3, \dots and variables x_1, x_2, x_3, \dots and if t_1 and t_2 are terms, then the following are also terms: $[t_1 + t_2]$, $[t_1 \cdot t_2]$, $!t_1$. The set of terms will be referred to as Tm . We also use a set $SLet$ of propositional variables, or sentence letters. These will usually be p_1, p_2, \dots . Formulas of the language L_n include all propositional variables and if ϕ, ψ are formulas, $i \in [n]$ and t is a term, then the following are also formulas of L_n : \perp , $\phi \rightarrow \psi$, $t :_i \phi$. The remaining propositional connectives, whenever needed, are treated as constructed from \rightarrow and \perp in the usual way. The operators \cdot , $+$ and $!$ are explained by the following axioms. Intuitively, \cdot applies a justification for a statement $A \rightarrow B$ to a justification for A and gives a justification for B . Using $+$ we can combine two justifications and have a justification for anything that can be justified by any of the two initial terms - much like the concatenation of two proofs. Finally, $!$ is a unary operator called the proof checker. Given a justification t for ϕ , $!t$ justifies the fact that t is a justification for ϕ .

Let $n \in \mathbb{N}$, $D, F \subseteq [n]$ and $V, C \subseteq [n]^2$. The logic $(J_{D,F,V,C}^n)_\emptyset$ is the logic with modus ponens as a derivation rule and the following axioms:

Propositional Axioms: Finitely many schemes of classical propositional logic;

Application: $s{:}_i(\phi \rightarrow \psi) \rightarrow (t{:}_i\phi \rightarrow [s \cdot t]{:}_i\psi)$;

Concatenation: $s{:}_i\phi \rightarrow [s + t]{:}_i\phi$, $s{:}_i\phi \rightarrow [t + s]{:}_i\phi$;

F-factivity: For every $i \in F$, $t{:}_i\phi \rightarrow \phi$;

D-consistency: For every $i \in D$, $t{:}_i\perp \rightarrow \perp$;

V-verification: For every $(i, j) \in V$, $t{:}_i\phi \rightarrow !t{:}_jt{:}_i\phi$;

C-conversion: For every $(i, j) \in C$, $t{:}_i\phi \rightarrow t{:}_j\phi$,

where in the above, ϕ and ψ are formulas in L_n , s, t are terms and $i, j \in [n]$. *F-factivity* and *D-consistency* are the usual factivity and consistency axioms for every agent in F and D respectively. Positive introspection is seen as a special case of *V-verification* - in this context, if agent i has positive introspection, then $(i, i) \in V$.

A constant specification for $J_{D,F,V,C}^n$ is any set

$$\mathcal{CS} \subseteq \{c{:}_iA \mid c \text{ is a constant, } A \text{ an axiom of } J_{D,F,V,C}^n \text{ and } i \in [n]\}.$$

We say that axiom A is justified by a constant c for agent i , when $c{:}_iA \in \mathcal{CS}$. A constant specification is: *axiomatically appropriate with respect to* $I \subseteq [n]$ if for every $i \in I$, each axiom is justified by at least one constant, *schematic* if every constant justifies only a certain number of schemes from the ones above (as a result, if c justifies A for i and B results from A and substitution, then c justifies B for i) and *schematically injective* if it is schematic and every constant justifies at most one scheme. Let $cl_n(\mathcal{CS})$ be the smallest set such that $\mathcal{CS} \subseteq cl_n(\mathcal{CS})$ and for every $t{:}_i\phi \in cl_n(\mathcal{CS})$, it is the case that for every $j \in [n]$, $!t{:}_jt{:}_i\phi \in cl_n(\mathcal{CS})$. $(J_{D,F,V,C}^n)\mathcal{CS}$ is $(J_{D,F,V,C}^n)_\emptyset + R4_{\mathcal{CS}}^n$, where $R4_{\mathcal{CS}}^n$ just outputs all elements of $cl_n(\mathcal{CS})$.

$(J_{D,F,V,C}^n)\mathcal{CS}$ is consistent: just map each formula to the propositional formula that is the result of removing all terms from the original one; then, all axioms are mapped to propositional tautologies and modus ponens preserves this mapping.

2.2 Semantics

We now introduce models for our logic. In the single-agent cases, M-models (introduced in [13, 9]) and F-models (introduced in [7, 14, 12]) are used (also in [3] for two-agent logics) and they are both important in the study of complexity issues. In this paper we are mostly interested in F-models, which we will usually just call models. These are essentially Kripke models with an additional machinery to accommodate justification terms.

Let $\mathcal{J} = (J_{D,F,V,C}^n)_{CS}$. Then, an F-model \mathcal{M} for \mathcal{J} is a quadruple $(W, (R_i)_{i \in [n]}, (\mathcal{A}_i)_{i \in [n]}, \mathcal{V})$, where $W \neq \emptyset$ is a set, for every $i \in [n]$, $R_i \subseteq W^2$ is a binary relation on W , $\mathcal{V} : SLet \rightarrow 2^W$ and for every $i \in [n]$, $\mathcal{A}_i : (Tm \times L_n) \rightarrow 2^W$. W is called the *universe* of \mathcal{M} and its elements are the worlds or states of the model. \mathcal{V} assigns a subset of W to each propositional variable, p , and \mathcal{A}_i assigns a subset of W to each pair of a justification term and a formula. Furthermore, $(\mathcal{A}_i)_{i \in [n]}$ will often be seen and referred to as $\mathcal{A} : [n] \times Tm \times L_n \rightarrow 2^W$ and \mathcal{A} is called an admissible evidence function. Additionally, \mathcal{A} must satisfy the following conditions:

Application closure: for any $i \in [n]$, formulas ϕ, ψ , and justification terms t, s ,

$$\mathcal{A}_i(s, \phi \rightarrow \psi) \cap \mathcal{A}_i(t, \phi) \subseteq \mathcal{A}_i(s \cdot t, \psi).$$

Sum closure: for any $i \in [n]$, formula ϕ , and justification terms t, s ,

$$\mathcal{A}_i(t, \phi) \cup \mathcal{A}_i(s, \phi) \subseteq \mathcal{A}_i(t + s, \phi).$$

CS -closure: for any formula $t:i\phi \in cl_n(CS)$, $\mathcal{A}_i(t, \phi) = W$.

V -Verification Closure: If $(i, j) \in V$, then $\mathcal{A}_i(t, \phi) \subseteq \mathcal{A}_j(!t, t:i\phi)$

C -Conversion Closure: If $(i, j) \in C$, then $\mathcal{A}_i(t, \phi) \subseteq \mathcal{A}_j(t, \phi)$

V -Distribution: for any formula ϕ , justification term t , $(i, j) \in V$ and $a, b \in W$, if aR_jb and $a \in \mathcal{A}_i(t, \phi)$, then $b \in \mathcal{A}_i(t, \phi)$.

The accessibility relations, R_i , must satisfy the following conditions:

- If $i \in F$, then R_i must be reflexive.
- If $i \in D$, then R_i must be serial ($\forall a \in W \exists b \in W aR_ib$).
- If $(i, j) \in V$, then for any $a, b, c \in W$, if aR_jbR_ic , we also have aR_ic .²
- For any $(i, j) \in C$, $R_j \subseteq R_i$.

Truth in the model is defined in the following way, given a state a :

- $M, a \not\models \perp$ and if p is a propositional variable, then $M, a \models p$ iff $a \in \mathcal{V}(p)$.
- If ϕ, ψ are formulas, then $M, a \models \phi \rightarrow \psi$ if and only if $M, a \models \psi$, or $M, a \not\models \phi$.
- If ϕ is a formula and t a term, then $M, a \models t:i\phi$ if and only if $a \in \mathcal{A}_i(t, \phi)$ and $M, b \models \phi$ for all $b \in W$ such that aR_ib .

A formula $\phi \in L_n$ is called satisfiable if there are some $M, a \models \phi$; we then say that \mathcal{M} satisfies ϕ in a . If CS is axiomatically appropriate with respect to D , then $(J_{D,F,V,C}^n)_{CS}$ is sound and complete with respect to its models; it is also sound and complete with respect to its models that have

² Thus, if i has positive introspection (i.e. $(i, i) \in V$), then R_i is transitive.

the *Strong Evidence Property*: $\mathcal{M}, a \models t:i\phi$ iff $a \in \mathcal{A}_I(t, \phi)$; furthermore, if ϕ is satisfiable, then it is satisfied by a model \mathcal{M} of at most $2^{|\phi|}$ states - and in fact, it is satisfied by a model \mathcal{M} of at most $2^{|\phi|}$ states that has the strong evidence property (see [3] for proofs of all the above that can be easily adjusted for this general case). A pair $(W, (R_i)_{i \in [n]})$ as above is called a frame for $(J_{D,F,V,C}^n)_{CS}$.

2.3 The *-calculus.

We present the *-calculi for $(J_{D,F,V,C}^n)_{CS}$. The *-calculi for the single-agent justification logics have proven to be an invaluable tool in the study of the complexity of these logics. This concept and results were adapted to the two-agent setting in [3] and here we extend them to the general multi-agent setting. Although the calculi have significant similarities to the ones of the single-agent justification logics, there are differences, notably that each calculus depends upon a frame and operates upon *-expressions (defined below) prefixed by states of the frame. A *-calculus was first introduced in [8], but its origins can be found in [13].

If t is a term, ϕ is a formula, and $i \in [n]$, then $*_i(t, \phi)$ is a star-expression (*-expression). Given a frame $\mathcal{F} = (W, (R_i)_{i \in [n]})$ and $V, C \subseteq [n]^2$ and constant specification CS , the $*_{CS}^{\mathcal{F}}(V, C)$ calculus on the frame \mathcal{F} is a calculus on *-expressions prefixed by worlds from W with the axioms and rules that are shown in figure 1.

Notice that the calculus rules correspond to the closure conditions of the admissible evidence functions. In fact and because of this, given a frame $\mathcal{F} = (W, R_1, R_2)$ and a set S of *-expressions prefixed by states of the frame, the function \mathcal{A} such that $\mathcal{A}_i(t, \phi) = \{w \in W \mid S \vdash_{*_{CS}^{\mathcal{F}}(V, C)} w *_i(t, \phi)\}$ is an admissible evidence function and in fact it is the minimal admissible evidence function such that for every $w *_i(t, \phi) \in S$, $w \in \mathcal{A}_i(t, \phi)$ in the sense that always $\mathcal{A}_i(t, \phi) \subseteq \mathcal{A}'_i(t, \phi)$ for any other admissible evidence function \mathcal{A}' such that for every $w *_i(t, \phi) \in S$, $w \in \mathcal{A}'_i(t, \phi)$. Therefore, given a frame $\mathcal{F} = (W, R_1, R_2)$ and two set \mathcal{T}, \mathcal{F} of *-expressions prefixed by states of the frame there is an admissible evidence function \mathcal{A} on \mathcal{F} such that for every $w *_i(t, \phi) \in \mathcal{T}$, $w \in \mathcal{A}_i(t, \phi)$ and for every $w *_i(t, \phi) \in \mathcal{F}$, $w \notin \mathcal{A}_i(t, \phi)$, if and only if there is no $f \in \mathcal{F}$ such that $\mathcal{T} \vdash_* f$. This observation yields the following.

Proposition 1. *For any³ constant specification CS , frame \mathcal{F} with universe W and $w \in W$, $(J_{D,F,V,C}^n)_{CS} \vdash t:i\phi \iff \vdash_{*_{CS}^{\mathcal{F}}(V, C)} w *_i(t, \phi)$.*

³ Note that we actually need an axiomatically appropriate constant specification to have completeness with respect to F-models, so we cannot immediately conclude

<p>*CS(\mathcal{F}) Axioms: $w *_i(t, \phi)$, where $t : i\phi \in cl_n(CS)$ and $w \in W$</p> <p>*App(\mathcal{F}):</p> $\frac{w *_i(s, \phi \rightarrow \psi) \quad w *_i(t, \phi)}{w *_i(s \cdot t, \psi)}$ <p>*Sum(\mathcal{F}):</p> $\frac{w *_i(t, \phi)}{w *_i(s + t, \phi)} \quad \frac{w *_i(s, \phi)}{w *_i(s + t, \phi)}$	<p>*V(\mathcal{F}): For any $(i, j) \in V$,</p> $\frac{w *_i(t, \phi)}{w *_j(!t, t : i\phi)}$ <p>*C(\mathcal{F}): For any $(i, j) \in C$,</p> $\frac{w *_i(t, \phi)}{w *_j(t, \phi)}$ <p>*V-Dis(\mathcal{F}): For any $(i, j) \in V$ and $(a, b) \in R_j$,</p> $\frac{a *_i(t, \phi)}{b *_i(t, \phi)}$
---	--

Fig. 1. The $*_{CS}^{\mathcal{F}}(V, C)$ -calculus: where $\mathcal{F} = (W, (R_i)_{i \in [n]})$ and for every $i \in [n]$

Proposition 2. *Let CS be a schematic constant specification in \mathbf{P} and $V, C \subseteq [n]^2$. Then, the following problem is in \mathbf{NP} : Given a finite frame $\mathcal{F} = (W, (R_i)_{i \in [n]})$ and a finite set S of $*$ -expressions prefixed by worlds from W , a formula ϕ , a term t , a $w \in W$ and $i \in [n]$, is it the case that*

$$S \vdash_{*_{CS}^{\mathcal{F}}(V, C)} w *_i(t, \phi)?$$

The proof of this proposition is very similar to the one that can be found in [10]. What is different here is the additional assignment of a state set to each node of the derivation tree, which does not change things a lot.

Proof. For this proof and every $j \in [n]$, let $f_j : 2^W \rightarrow 2^W$, s.t. for every $X \subseteq W$, $f_j(X) = \{y \in W \mid \exists x \in X, (j, j') \in V \ x R_{j'} y\} \cup X$.

- Nondeterministically construct a rooted tree with pairs of the form (j, s) , where $j \in [n]$ and s is a subterm of t , as nodes, such that (i, t) is the root and the following conditions are met. Node (j, s) can be the parent of (j_1, s_1) or of both (j_1, s_1) and (j_2, s_2) as long as there is a rule $\frac{w *_j(s, \phi_3)}{w *_j(s, \phi_3)}$ or $\frac{w *_j(s, \phi_3) \quad w *_j(s, \phi_3)}{w *_j(s, \phi_3)}$, respectively, of the $*$ -calculus and as long as $(j_1, s_1) \neq (j, s) \neq (j_2, s_2)$. To keep

with this result for *any* constant specification. Nevertheless, proposition 1 holds, but to prove it we need to introduce M-models, but we do not use M-models anywhere else. Thus, the reader can see [8, 10], or [3] for a proof of proposition 1 for all constant specifications.

this structure a tree, we can ensure at this step that there are no cycles, which would correspond to consecutive applications of $*C(\mathcal{F})$ and which would be redundant.

- Nondeterministically assign to each leaf, (j, l) , either
 - some formula ψ and the closure under f_j of some set $W' \subseteq W$, s.t. for every $w \in W'$, $w *_j(l, \psi) \in S$ or,
 - as long as l is of the form $\underbrace{! \cdots !}_k c$, where c a constant, $k \in \mathbb{N}$, then we can also assign some $\underbrace{! \cdots !}_{k-1} c_{i_k} \cdots ! c_{i_2} c_{i_1} A$ and $W' = W$, where A an axiom scheme, s.t. $c_{i_1} A \in \mathcal{CS}$.
- If for some node $\nu = (j, s)$ all its children, say $\nu_1 = (j_1, s_1), \nu_2 = (j_2, s_2)$ have been assigned some scheme or formula P_1, P_2 and world sets V_1, V_2 , assign to ν some scheme or formula P , such that P_1, P_2 can be unified to P'_1, P'_2 such that $\frac{w *_{j_1}(s_1, P'_1) \quad w *_{j_2}(s_2, P'_2)}{w *_j(s, P)}$ is a rule in the $*_{\mathcal{CS}}^{\mathcal{F}}(V, C)$ -calculus and world set V , where V is the closure of $V_1 \cap V_2$ under f_j . Apply this step until the root of the tree has been assigned some scheme or formula and a W' subset of W .
- Unify ϕ with the formula assigned to (i, t) and verify that $w \in W'$

If some step is impossible, the algorithm rejects. Otherwise, it accepts. Using efficient representations of schemes using DAGs and Robinson's unification algorithm, the algorithm runs in polynomial time. We can see that as the tree is constructed, if (i, s) is assigned scheme P and set V , then the construction effectively describes a valid derivation of any expression of the form $v *_i(s, \psi)$, where $v \in V$ and ψ an instance of P . Therefore, if the algorithm accepts, there exists a valid $*_{\mathcal{CS}}^{\mathcal{F}}(V, C)$ -calculus derivation of $w *_i(t, \phi)$. On the other hand if there is some $*_{\mathcal{CS}}^{\mathcal{F}}(V, C)$ -calculus derivation for $w *_i(t, \phi)$ from S , then the algorithm in the first two steps can essentially describe this derivation by producing the derivation tree and the formulas/schemes by which the derivation starts. Therefore, the algorithm accepts if and only if there is a $*_{\mathcal{CS}}^{\mathcal{F}}(V, C)$ -calculus derivation for $w *_i(t, \phi)$ from S . See [8] and [10] for a more detailed analysis. \square

The number of nondeterministic choices made by the algorithm in the proof of proposition 2 is bounded by $|t| + |S'|$, where $S' = \{*_j(s, \psi) \mid \exists w *_j(s, \psi) \in S\}$. Therefore, if there is some formula ψ such that $t *_i \phi$ is a subformula of ψ and for every $*_j(s, \psi') \in S'$, $s *_j \psi'$ is a subformula of ψ , then $2|\psi| \geq |t| + |S'|$ and therefore we can simulate all nondeterministic choices in time $2^{O(|\psi|)}$. Thus the algorithm can be turned into a deterministic one running in time $2^{O(|\phi|)} \cdot O(|W|^2)$. This observation, the fact that a

satisfiable ϕ can be satisfied by a model of at most $2^{|\phi|}$ states (see the previous subsection) and the previous two propositions give the following results:

Corollary 1. *Let $\mathcal{J} = (J_{D,F,V,C}^n)_{\mathcal{CS}}$, where $\mathcal{CS} \in \mathbf{P}$ is schematic. Then,*

1. *Deciding for $t;_i\phi$ that $\mathcal{J} \vdash t;_i\phi$ is in NP.*
2. *If \mathcal{CS} is axiomatically appropriate with respect to D , then the satisfiability problem for \mathcal{J} is in NEXP.*

Additionally notice that if the term t has no $+$, \mathcal{CS} is schematically injective and $S = \emptyset$, we have essentially eliminated nondeterministic choices from the procedure above. Thus, we conclude (for the original result, see [6]):

Corollary 2. *Let $\mathcal{J} = (J_{D,F,V,C}^n)_{\mathcal{CS}}$, where $\mathcal{CS} \in \mathbf{P}$ is schematically injective. Then, deciding for $t;_i\phi$, where t has no ‘+’, that $\mathcal{J} \vdash t;_i\phi$ is in \mathbf{P} .*

3 Tableaux

In this section we give a general tableau procedure for every logic which varies according to each logic’s parameters. We can then use the tableau for a particular logic and make observations on its complexity, as we do in the following section. A version of the tableau which is more efficient for some cases follows after. To develop the tableau procedure we need to examine the relations on the agents more carefully than we have so far. For this section and the following one fix some $J = (J_{D,F,V,C}^n)_{\mathcal{CS}}$ and we assume \mathcal{CS} is axiomatically appropriate with respect to D and schematic.

3.1 A Closer Look on the Agents and their Interactions

If A_1, \dots, A_k are binary relations on the same set, then $A_1 \cdots A_k$ is the binary relation on the same set, such that $x A_1 \cdots A_k y$ if and only if there are x_1, \dots, x_k in the set, such that $x = x_1 A_1 x_2 A_2 \cdots A_{k-1} x_k A_k y$. If A is a binary relation, then A^* is the reflexive, transitive closure of A ; if A is a set (but not a set of pairs), then A^* is the set of strings from A . We also use the following relation on strings: $a \sqsubseteq b$ iff there is some string c such that $ac = b$. We define the following subsets of and relations on $[n]$.

$S = S(J) = \{i \in [n] \mid (\exists j \in D \cup F) i C^* j\}$ attempts to capture exactly those agents that require a serial accessibility relation; from now and on these agents monopolize our attention;

$R = R(J) = \{i \in [n] | (\exists j \in F) iC^*j\}$ attempts to capture exactly those agents that require a reflexive accessibility relation;
 $C_F = C \cup \{(i, j) \in V | i \in R, j \in S\}$ notice that if iC_Fj , and xR_jy , then xR_iy ;⁴
 $Q = (V|_S \cup C|_S)^*$: if iQj , then i 's justifications somehow affect j 's justifications; conversely, j 's accessibility relations somehow affect i 's accessibility relations;
 $\equiv_C = C_F^* \cap (C_F^*)^{-1}$: $i \equiv_C j$ if and only if iC_F^*j and jC_F^*i ; we can easily see that \equiv_C is an equivalence relation and that if $i \equiv_C j$, then i and j have the same accessibility relations; similarly,

For the above equivalence relations we can define equivalence classes on S , $P_C = \{L_1, \dots, L_{k_C}\}$. $\chi(i)$ is the equivalence class $L \in P_C$ s.t. $i \in L$.

We can define relations $<_C$ and $<_{VC}$ on P_C in the following way: $P_1 \leq_C P_2$ iff $\exists x \in P_1 \exists y \in P_2$ s.t. xC_F^*y and $P_1 \leq_{VC} P_2$ iff $\exists x \in P_1 \exists y \in P_2$ s.t. xQy . Also, $P_1 <_C P_2$ iff $P_1 \leq_C P_2$ and $P_1 \not\leq_C P_2$ and similarly for $P_1 <_{VC} P_2$. Then, define $P_1 \leq_V P_2$ iff there are $x \in P_1$ and $y \in P_2$ s.t. $xQVQy$, that is, there are $x_1, x_2 \in S$, where xQx_1Vx_2Qy . $P_1 <_V P_2$ iff $P_1 \leq_V P_2$ and $P_2 \not\leq_V P_1$.

Let $i \in [n]$. Then, $M_C(i) = M_C(\chi(i)) = \{L \in P_C | \chi(i) \leq_C L \text{ and } \nexists L' \in P_C \text{ s.t. } L <_C L'\}$.

3.2 The Tableau Procedure.

The formulas used in the tableau will have the form $\emptyset.\sigma \ s \ \beta\psi$, where $\psi \in L_n$ or is a $*$ -expression, $\sigma \in P_C^*$ (the world prefixes are strings of equivalence classes of agents), β is (either the empty string or) of the form $\Box_i \Box_j \dots \Box_k, i, j, \dots, k \in [n]$, and $s \in \{T, F\}$. Furthermore, $\emptyset.\sigma$ will be called a world-prefix or state-prefix, s a truth-prefix and world prefixes will be denoted as $\emptyset.s_1.s_2 \dots s_k$, instead of $\emptyset.s_1s_2 \dots s_k$, where for all $x \in [k]$, $s_x \in P_C$.

A tableau branch is a set of formulas of the form $\sigma \ s \ \beta\psi$, as above. A branch is complete if it is closed under the tableau rules (they follow). It is propositionally closed if $\sigma \ T \ \beta\psi$ and $\sigma \ F \ \beta\psi$ are both in the branch. We say that a tableau branch is constructed by the tableau rules from ϕ , if it is a closure of $\{\emptyset \ T \ \phi\}$ under the rules.

For every $i, j \in S$, $i \in N(j)$ if there are some $i', i'' \in S \setminus R$ such that $i' \equiv_{VC} i'', i''VC_F^*j$ and $\chi(i) \in M_C(i')$.

⁴ The F in C_F is to indicate that C_F is a variation of C influenced by the agents in F , not that it is available with other subscripts

The tableau rules will include certain classical rules to cover propositional cases of formulas, as well as the ones that follow:

$$\begin{array}{ll}
\frac{\sigma \ T \ t{:}_i\psi}{\sigma \ T \ *_i(t, \psi)} \text{TRB} & \text{if } i \in S; \\
\sigma \ T \ \Box_i\psi & \\
\frac{\sigma \ T \ t{:}_i\psi}{\sigma.\chi(j) \ F \ \perp} \text{TRD} & \text{if } i \in S, \chi(j) \in M_C(\chi(i)), \text{ and } j \notin R; \\
\sigma \ T \ t{:}_i\psi & \\
\frac{\sigma \ T \ t{:}_i\psi}{\sigma \ T \ *_i(t, \psi)} \text{TR} & \text{if } i \notin S; \\
\sigma \ F \ t{:}_i\psi & \\
\frac{\sigma \ F \ t{:}_i\psi}{\sigma \ F \ *_i(t, \psi)} \text{FA} & \\
\frac{\sigma.\chi(j) \ F \ \perp}{\sigma.\chi(j).\chi(i) \ F \ \perp} \text{S} & \text{if } i \in N(j); \\
\sigma \ T \ \Box_i\psi & \\
\frac{\sigma \ T \ \Box_i\psi}{\sigma.\chi(i) \ T \ \psi} \text{SB} & \text{if } \sigma.\chi(i) \text{ has already appeared}; \\
\sigma \ T \ \Box_i\psi & \\
\frac{\sigma \ T \ \Box_i\psi}{\sigma \ T \ \psi} \text{FB} & \text{if } i \in F; \\
\sigma \ T \ \Box_i\psi & \\
\frac{\sigma \ T \ \Box_i\psi}{\sigma \ T \ \Box_j\psi} \text{C} & \text{if } iCj; \\
\sigma \ T \ \Box_i\psi & \\
\frac{\sigma \ T \ \Box_i\psi}{\sigma \ T \ \Box_j\Box_i\psi} \text{V} & \text{if } iVj.
\end{array}$$

We do not explicitly mention it anywhere else, but of course, we need a set of rules to cover propositional cases as well. In particular we can use

$$\frac{\sigma \ T \ \psi \rightarrow \psi'}{\sigma \ F \ \psi \quad | \quad \sigma \ T \ \psi'} \quad \text{and} \quad \frac{\sigma \ F \ \psi \rightarrow \psi'}{\sigma \ T \ \psi \quad | \quad \sigma \ F \ \psi'}$$

The separator $|$ indicates a nondeterministic choice between the two options it separates.

If b is a tableau branch, then⁵ $W(b) = \{\sigma \in \emptyset.P_C^* \mid \text{there is some } \sigma \ a \in b\}$. Let $(R_i)_{i \in [n]}$ be such that for every $i \in [n]$,

$$R_i = \{(\sigma, \sigma.\chi(i)) \in (W(b))^2\} \cup \{(w, w) \in (W(b))^2 \mid i \in F\}$$

⁵ $\emptyset.P_C^*$ here and wherever else it may appear is the set $\{\emptyset.x \mid x \in P_C^*\}$.

then $\mathcal{F}(b) = (W(b), (R'_i)_{i \in [n]})$, where $(R'_i)_{i \in [n]}$ is the closure of $(R_i)_{i \in [n]}$ under the conditions of frames for the accessibility relations, except for seriality. $(R'_i)_{i \in [n]}$ is constructed in the following way: for every $i \in [n]$, let $R_i^0 = R_i$ and for every $k \in \mathbb{N} \cup \{0\}$,

$$R_i^{k+1} = R_i^k \cup \bigcup_{(i,j) \in C} R_j^k \cup \bigcup_{(i,j) \in V} \{(a,b) \in (W(b))^2 \mid \exists(a,c) \in R_j^k, (c,b) \in R_i^k\}$$

and then, $\mathcal{F}(b) = (W(b), (\bigcup_{k \in \mathbb{N}} R_i^k)_{i \in [n]})$.

Finally, let $T(b) = \{\sigma *_i(t, \psi) \mid \sigma T *_i(t, \psi) \text{ appears in } b\}$ and $F(b) = \{\sigma *_i(t, \psi) \mid \sigma F *_i(t, \psi) \text{ appears in } b\}$.

A branch b of the tableau is rejecting when it is propositionally closed or there is some $f \in F(b)$ such that $T(b) \vdash_{*_{\mathcal{CS}}^{(b)}(V,C)} f$. Otherwise it is an accepting branch.

By induction on the construction of $\mathcal{F}(b)$, it is not hard to see that for every $(\sigma, \tau.\chi(j)) \in R_i$, it must be the case that iC_F^*j or that $i \in R$ and $\sigma = \tau.\chi(j)$. By induction on the frame construction we can see that if $\sigma T \Box_i \phi$ appears in b and $\sigma R_i \tau$, then $\tau T \phi$ appears in b .

Proposition 3. *If there is a complete accepting tableau branch $b \ni \emptyset T \phi$, then the formula ϕ is satisfiable by a model for J .*

Proof. Let $\mathcal{M} = (W, (R_i)_{i \in [n]}, \mathcal{A}, \mathcal{V})$, where $(W, (R_i)_{i \in [n]}) = \mathcal{F}(b)$, $\mathcal{V}(p) = \{w \in W \mid w T p \in b\}$, and $\mathcal{A}_i(t, \psi) = \{w \in W \mid T(b) \vdash_{*_{\mathcal{CS}}^{(b)}(V,C)} w *_i(t, \psi)\}$.

Let $\mathcal{M}' = (W, (R'_i)_{i \in [n]}, (\mathcal{A}_i)_{i \in [n]}, \mathcal{V})$, where for every $i \in [n]$, if $i \in S$, then $R'_i = R_i \cup \{(a, a) \in W^2 \mid \exists j \in S \text{ s.t. } iC_F^*j, \nexists(a, b) \in R_j\}$ and $R'_i = R_i$, otherwise. \mathcal{M}' is an F-model for J : $(\mathcal{A}_i)_{i \in [n]}$ easily satisfy the appropriate conditions, as the extra pairs of the accessibility relations do not affect the $*$ -calculus derivation, and we can prove the same for $(R'_i)_{i \in [n]}$. If $aR'_i bR'_j c$ and jVi , if $(a, b) \in R'_i \setminus R_i$, then $a = b$ and thus $aR'_i c$. If $(a, b) \in R_i$, then, from rule S, there must be some $(b, c') \in R_j$, so $(b, c) \in R_j$ and thus, $(a, c) \in R_j$. If $(a, b) \in R'_i$ and jCi , then, trivially, whether $(a, b) \in R_i$ or not, $(a, b) \in R'_j$.

By induction on χ , we prove that for every formula χ and $a \in W$, if $a T \chi \in b$, then $\mathcal{M}', a \models \chi$ and if $a F \chi \in b$, then $\mathcal{M}', a \not\models \chi$. Propositional cases are easy. If $\chi = t;_i \psi$ and $a F \chi \in b$, then $a \notin \mathcal{A}_i(t, \psi)$, so $\mathcal{M}', a \not\models \chi$. On the other hand, if $a T t;_i \psi \in b$, then $a \in \mathcal{A}_i(t, \psi)$ and by rule TrD, for every $j \in S$ such that iC_F^*j , there is some $(a, b) \in R_j$. Therefore, for every $(a, b) \in R'_j$, it is the case that $(a, b) \in R_j$, so by rule TrB, a previous observation about formulas of the form $w T \Box_i \alpha$, and the inductive hypothesis, for every $(a, b) \in R_i$, $\mathcal{M}', b \models \psi$ and therefore, $\mathcal{M}', a \models t;_i \psi$. \square

Now we can prove the following proposition.

Proposition 4. *Let $\phi \in L_n$. ϕ is $(J_{D,F,V,C}^n)_{CS}$ -satisfiable if and only if there is a complete tableau branch b that is produced from $\emptyset \ T \ \phi$, such that*

- for all σ, α , not both $\sigma \ T \ \alpha$ and $\sigma \ F \ \alpha$ appear in b and
- For any $\beta \in F(b)$, $T(b) \not\models_{*_{\mathcal{F}_{CS}(V,C)}} \beta$.

Proof. The “if” direction was handled by proposition 3. We will prove the “only if” in the following. Let \mathcal{M} be an F-model, $(W, (R_i)_{i \in [n]}, (\mathcal{A}_i)_{i \in [n]}, \mathcal{V})$ that has the strong evidence property and a state $s \in W$ such that $\mathcal{M}, s \models \phi$. Furthermore, fix some $\cdot^{\mathcal{M}} : \emptyset.P_C^* \rightarrow W$, such that $\emptyset^{\mathcal{M}} = s$ and the following conditions are met. For any $\sigma.\chi(i) \in P_C^*$, $(\emptyset.\sigma.\chi(i))^{\mathcal{M}}$ is some element of W s.t. $((\emptyset.\sigma)^{\mathcal{M}}, (\emptyset.\sigma.\chi(i))^{\mathcal{M}}) \in R_i$.

Let $L_n^\square = \{\Box_{i_1} \dots \Box_{i_k} \phi \mid \phi \in L_n, k \in \mathbb{N}, i_1, \dots, i_k \in [n]\}$. Given a state a of the model, and $\Box_i \psi \in sub_\square(\phi)$, $\mathcal{M}, a \models \Box_i \psi$ has the usual, modal interpretation, $\mathcal{M}, a \models \Box_i \psi$ iff for every $(a, b) \in R_i$, $\mathcal{M}, b \models \psi$

We can see in a straightforward way and by induction on the tableau derivation that there is a branch, such that if $\sigma \ T \ \psi$ appears in the branch and $\psi \in L_n^\square$, then $\mathcal{M}, \sigma^{\mathcal{M}} \models \psi$, if $\sigma \ F \ \psi$ appears in the branch and $\psi \in L_n^\square$, then $\mathcal{M}, \sigma^{\mathcal{M}} \not\models \psi$, if $\sigma \ T \ *_i(t, \psi)$ appears in the branch, then $\sigma^{\mathcal{M}} \in \mathcal{A}_i(t, \psi)$ and if $\sigma \ F \ *_i(t, \psi)$ appears in the branch, then $\sigma^{\mathcal{M}} \notin \mathcal{A}_i(t, \psi)$. The proposition follows. \square

3.3 An Improved Tableau

By taking a closer look at the interactions between the agents we can further improve the efficiency of our tableau procedure and we do that in this section. We use this improvement to prove an upper bound on the complexity of a general class of logics. We need the following definitions and lemma 1, which is a generalization of a result from [1] and has appeared in simpler forms in [3].

First we define an additional equivalence relation: $\equiv_{VC} = Q \cap Q^{-1}$. As an equivalence relation, this one too gives equivalence classes on S and they are $P_{VC} = \{P_1, \dots, P_{k_{VC}}\}$. Notice that as $C^* \subseteq (V \cup C)^*$, $\equiv_C \subseteq \equiv_{VC}$ and therefore P_C is a refinement of P_{VC} . We’ll call $P = P_{VC}$. Furthermore, notice that for any $L \in P$, either $\exists x, y \in L$ s.t. xVy , or $L \in P_C$. In the first case, L will be called a *V-class* of agents and in the second case it will be called a *C-class* of agents. For each agent $i \in [n]$, $P(i)$ will be the equivalence class $L \in P$ s.t. $i \in L$ and To help keep the relationship between the sets of equivalence classes in mind, notice that

$i \in \chi(i) \subseteq P(i) \subseteq S \subseteq [n]$. Furthermore, we can extend relations $<_C$ and $<_{VC}$ on P in the same way they were defined on P_C .

Lemma 1. *Let $\mathcal{M} = (W, (R_i)_{i \in [n]}, (\mathcal{A}_i)_{i \in [n]}, \mathcal{V})$ be a $(J_{D,F,V,C}^n)$ CS F -model of at most $2^{|\phi|}$ states, $P_a \in P$ a V -class of agents, and $u \in W$. Then, there are states of W , $(a_i)_{i \in P_a}$, such that*

1. *For any $i \in P_a$, uR_ia_i .*
2. *For any $i, j \in P_a$, $v, b \in W$, if a_i, bR_jv , then bR_ja_j .*

$(a_i)_{i \in P_a}$ will be called a P_a -cluster for u .

Proof. For this proof we need to define the following. Let $i \in [n]$, $w, v \in W$. An E_V -path ending at i (and starting at i') from w to v is a finite sequence v_1, \dots, v_{k+1} , such that for some $j_1, \dots, j_k \in [n]$, $E_1, \dots, E_{k-1} \in \{C^{-1}, V^{-1}\}$, where for some $j \in [k-1]$ $E_j = V^{-1}$ and $j_k = i$ (and $j_1 = i'$), for every $a \in [k-1]$, $j_a E_a j_{a+1}$ and if $E_a = C^{-1}$, then $v_{a+1} = v_{a+2}$, while if $E_a = V^{-1}$, then $v_{a+1} R_{j_{a+1}} v_{a+2}$ and $v_1 = w$, $v_k = v$, $v_1 R_1 v_2$. The E_V -path covers a set $s \subseteq [n]$ if $\{j_1, \dots, j_k\} = s$. For this path and $a \in [k]$, v_{a+1} is a j_a -state. Notice that if there is an E_V path ending at i from w to v and some $j \in s$ and $z \in W$ such that the path covers s and zR_jw , it must also be the case that w, zR_iv .

Let $p : [m] \rightarrow P_a$ be such that $m \in \mathbb{N}$, $p[[m]] = P_a$ and for every $i+1 \in [m]$, either $p(i+1)Cp(i)$ or $p(i+1)Vp(i)$ and there is some $i+1 \in [m]$ such that $p(i+1)Vp(i)$. For any $s \in W$, $x \in \mathbb{N}$ let $b_0(s), b_1(s), b_2(s), \dots, b_m(s)$ be the following: $b_0(s) = s$, for all $k \in [m]$, $b_1(s)$ will be such that there is an E_V path ending at $p(1)$ from s to $b_1(s)$ and covering P_a and if $k > 1$, $b_k(s)$ is such that $b_0(s), b_1(s), b_2(s), \dots, b_k(s)$ is an E_V path ending at $p(k)$. Let $(b_i^x)_{i \in [m], x \in \mathbb{N}}, (a_i^x)_{i \in [m], x \in \mathbb{N}}$ be defined in the following way. For every $i \in [m]$, $b_i^0 = b_i(u)$ and for every $x \in \mathbb{N}$, $a_i^x = b_i(b_m^x)$. Finally, for $0 < x \in \mathbb{N}$, $(b_i^x)_{i \in [m]}$ is defined in the following way. If there are some $b_x, v \in W$, $i, j \in P_a$, such that $b_x R_j v$, $a_i^{x-1} R_j v$ and not $b_x R_j a_j^{x-1}$, then for all $i \in P_a$, $b_i^x = b_i(v)$. Otherwise, $(b_i^x)_{i \in P_a} = (a_i^x)_{i \in P_a}$. By induction on x , we can see that for every $x, y \in \mathbb{N}$, $i \in P_a$, if $y \geq x$, then $b_x R_i b_i^y, a_i^y$. Since the model has a finite number of states, there is some $x \in \mathbb{N}$ such that for every $y \geq x$, $(b_i^y)_{i \in P_a} = (a_i^y)_{i \in P_a}$. Therefore, we can pick appropriate $(a_i)_{i \in P_a}$ among $(a_i^k)_{i \in P_a}$ that satisfy conditions 1, 2. \square

We recursively define relation \rightarrow on S^* :

- if $jC_F i$ then $i \rightarrow j$;
- if $jV i$, then $ij \rightarrow j$;

– If $\beta \rightarrow \delta$, then $\alpha\beta\gamma \rightarrow \alpha\delta\gamma$.

\rightarrow^* is the reflexive, transitive closure of \rightarrow . \rightarrow^* tries to capture the closure of the conditions on the accessibility relations of a frame. This is made explicit by observing that if for some frame $(W, (R_i)_{i \in [n]})$, $aR_{i_1}R_{i_2} \cdots R_{i_k}b$ and $i_1i_2 \cdots i_k \rightarrow^* j_1j_2 \cdots j_l$, then $aR_{j_1}R_{j_2} \cdots R_{j_l}b$. Furthermore, if, in addition, $l = k$, then for every $r \in [k]$, $j_r C_F^* i_r$. For every agent $i \in S$, we introduce a new agent, \bar{i} and we extend \rightarrow^* , so that when $P(i)$ a V-class, then $\bar{i} \rightarrow^* \chi$ for every $\chi \in S^*$ such that $\chi \rightarrow^* i$. For each $L \in P_C$, we fix some $i \in P_C$ and $\bar{L} = \bar{i}$. Furthermore, if $xy \in P_C \cup S^*$, then $\bar{x}\bar{y} = \bar{x}\bar{y}$. This extended definition of \rightarrow^* tries to capture the closure of the conditions on the accessibility relations of a frame like the ones that will result from a tableau procedure as defined in the following.

Let $L \in P$ and σ be a finite string of elements from P_C . Then, L is *visible* from $\emptyset.\sigma$ if and only if there is some $\chi(i) \subseteq L$, some $\chi \in P_C^*$ and some $\alpha \in S^*$ such that $\sigma = \tau.\chi(i).\chi$ and $\bar{\chi}\alpha \rightarrow^* i$; $\tau.\chi(i)$ is then called the L -view from σ . Notice that there is a similarity between this definition and the statement of lemma 1 - this will be made explicit later on.

Then we adjust the tableau by altering rules TrD and S and introduce rule SVB:

$$\begin{array}{ll} \frac{\sigma \ T \ t:i\psi}{\sigma.\chi(j) \ F \ \perp} \text{ TrD} & \text{if } i \in S, \chi(j) \in M_C(\chi(i)), j \notin R, \text{ and } P(j) \text{ is not a V-class visible from } \sigma; \\ \frac{\sigma.\chi(j) \ F \ \perp}{\sigma.\chi(j).\chi(i) \ F \ \perp} \text{ S} & \text{if } i \in N(j) \text{ and it is not the case that } P(i) \text{ is a V-class visible from } \sigma.\chi(j); \\ \frac{\sigma \ T \ \Box_i\psi}{\tau.\chi(i) \ T \ \psi} \text{ SVB} & \text{if } P(i) \text{ a V-class, visible from } \sigma, \tau.\chi(j) \text{ is the } P(i)\text{-view from } \sigma \text{ and } \tau.\chi(i) \text{ has already appeared in the tableau.} \end{array}$$

Then we have to redefine the frame $\mathcal{F}(b)$. Let $(R_i)_{i \in [n]}$ be such that for every $i \in [n]$,

$$R_i = \{(\sigma, \sigma.\chi(i)) \in (W(b))^2\} \cup \{(w, w) \in (W(b))^2 \mid i \in F\}$$

\(\cup\)

$$\{(\sigma, \tau.\chi(i)) \in (W(b))^2 \mid P(i) \text{ a V-class, } \tau.\chi(j) \text{ the } P(i)\text{-view from } \sigma\}$$

and $\mathcal{F}(b) = (W(b), (R'_i)_{i \in [n]})$, where $(R'_i)_{i \in [n]}$ is the closure of $(R_i)_{i \in [n]}$ as it was defined before.

Proposition 3 and its proof remain the same. To prove proposition 4 for this version of the rules, follow the same proof, but for every $a \in W$ and V-class L fix some L -cluster for a and if $P(i)$ is a V-class, then for

$(a_j)_{j \in L}$, the fixed L -cluster for $\sigma^{\mathcal{M}}$, $(\sigma.\chi(i))^{\mathcal{M}} = a_i$ - and observe that if $P(i)$ a V -class, visible from σ and $\tau.\chi(j)$ is the $P(i)$ -view from σ , then in model \mathcal{M} there is some v such that $\sigma^{\mathcal{M}}, (\tau.\chi(j))^{\mathcal{M}} R_i v$, which by the definition of clusters in turn means that $\sigma^{\mathcal{M}} R_j (\tau.\chi(j))^{\mathcal{M}}$. The remaining proof is the same.

Notice that if for every appearing world-prefix $\sigma.\chi(i)$, i is always in the same V -class L , then all prefixes are of the form $\emptyset.\chi(j)$, where $j \in L$. In that case we can simplify the box rules and in particular just ignore rule V and end up with the following result.

Corollary 3. *When there is some V -class L such that for every $i \in S \setminus R$ there is some $i' \in L$ such that $i C_F^* i'$, then $(J_{D,F,V,C}^n)_{CS}$ -satisfiability is in Σ_2^P .*

4 Complexity Jumps

In this section we look into some more specific cases of multi-agent justification logics and demonstrate certain jumps in the complexity of the satisfiability problem for these logics. We first revisit the two-agent logics from [3]. Like in the previous sections, we assume our constant specifications are schematic and axiomatically appropriate (and in P for upper bounds).

Our definition here of $(J_{D,F,V,C}^n)_{CS}$ allows for more two-agent logics than the ones that were studied in [3]. It is not hard, though, to extend those results to all two-agent cases of $(J_{D,F,V,C}^n)_{CS}$: when there are $\{i, j\} = [2]$, $i \in D \setminus F$, $\emptyset \neq V \subseteq \{(i, j), (j, j)\}$, $(i, j) \in C$, and $(j, i) \notin C$, then $(J_{D,F,V,C}^2)_{CS}$ -satisfiability is PSPACE-complete; otherwise it is in Σ_2^P (see [3]).

We will further examine the following two cases. $\mathcal{J}_1, \mathcal{J}_2$ are defined in the following way. $n_1 = n_2 = 3$; $D_1 = D_2 = \{1, 2\}$; $F_1 = F_2 = \emptyset$; $V_1 = \emptyset$, $V_2 = \{(3, 3)\}$; $C_1 = C_2 = \{(3, 1), (3, 2)\}$; finally, for $i \in [2]$, $\mathcal{J}_i = (J_{D_i, F_i, V_i, C_i}^{n_i})_{CS_i}$, where CS_i is some axiomatically appropriate and schematic constant specification.

By an adjustment of the reductions in [2], as it was done in [3], it is not hard to prove that \mathcal{J}_1 is PSPACE-hard and \mathcal{J}_2 is EXP-hard.⁶ Notice that

⁶ \mathcal{J}_1 would correspond to what is defined in [3] as $D_2 \oplus_{\subseteq} K$ and \mathcal{J}_2 to $D_2 \oplus_{\subseteq} D4$. Then we can pick a justification variable x and we can either use the same reductions and substitute \Box_i by $x_{:,i}$ or we can just translate each diamond-free fragment to the corresponding justification logic in the same way. It is not hard to see then that the original modal formula behaves exactly the same way as the result of its translation with respect to satisfiability - just consider F -models where always $\mathcal{A}_i(t, \phi) = W$.

the way we prove PSPACE-hardness for \mathcal{J}_1 is different in character from the way we prove the same result for the two-agent logics in [3]. For \mathcal{J}_1 we use the way the tableau prefixes for it branch, while for $(\text{JD4} \times_C \text{JD})_{\text{CS}}$ the prefixes do not branch, but they increase to exponential size.

In fact, we can see that \mathcal{J}_1 is PSPACE-complete, while \mathcal{J}_2 is EXP-complete. The respective tableau rules as they turn out for each logic are - notice that neither logic has any \leq_C -maximal V-classes:

The rules for \mathcal{J}_1 are:

$$\begin{array}{ccc}
\frac{\sigma \ T \ t{:}_i\psi}{\sigma \ T \ *_i(t, \psi)} \text{TrB} & \frac{\sigma \ T \ t{:}_1\psi}{\sigma.\{1\} \ F \ \perp} \text{TrD} & \frac{\sigma \ T \ t{:}_2\psi}{\sigma.\{2\} \ F \ \perp} \text{TrD} \\
\\
\frac{\sigma \ T \ t{:}_3\psi}{\sigma.\{1\} \ F \ \perp} \text{TrD} & \frac{\sigma \ F \ t{:}_i\psi}{\sigma \ F \ *_i(t, \psi)} \text{FA} & \frac{\sigma \ T \ \Box_i\psi}{\sigma.\{i\} \ T \ \psi} \text{SB} \\
\sigma.\{2\} \ F \ \perp & & \text{if } \sigma.\{i\} \text{ has already appeared;} \\
\\
\frac{\sigma \ T \ \Box_3\psi}{\sigma \ T \ \Box_1\psi} \text{C} & & \\
\sigma \ T \ \Box_2\psi & &
\end{array}$$

Notice that the maximum length of a world prefix is at most $|\phi|$, since the depth (nesting of terms) of the formulas decrease whenever we move from σ to $\sigma.\{i\}$. Also notice that when we run the $*$ -calculus, there is no use for rule $*V\text{-Dis}$, so we can simply run the calculus on one world-prefix at the time, without needing the whole frame. Therefore, we can turn the tableau into an alternating polynomial time procedure, which uses a nondeterministic choice when the tableau would make a nondeterministic choice (when we apply the propositional rules) and uses a universal choice to choose whether to increase prefix σ to $\sigma.\{1\}$ or to $\sigma.\{2\}$. This means that \mathcal{J}_1 -satisfiability is PSPACE-complete.

The rules for \mathcal{J}_2 are the same with the addition of:

$$\begin{array}{ccc}
\frac{\sigma \ T \ \Box_3\psi}{\sigma \ T \ \Box_3\Box_3\psi} \text{V} & & \frac{\sigma.\{i\} \ F \ \perp}{\sigma.\{i\}.\{1\} \ F \ \perp} \text{S} \\
& & \sigma.\{i\}.\{2\} \ F \ \perp
\end{array}$$

For the tableau procedure of \mathcal{J}_2 we have no such bound on the size of the largest world-prefix, so we cannot have an alternating polynomial time procedure. As before, though, the $*$ -calculus does not use rule $*V\text{-Dis}$, so again we can run the calculus on one world-prefix at the time. Furthermore, for every prefix w , $|\{w \ a \in b\}|$ is polynomially bounded (observe that we do not need more than two boxes in front of any formula),

so in turn we have an alternating polynomial space procedure. Therefore, \mathcal{J}_2 -satisfiability is EXP-complete.

References

1. Antonis Achilleos. A complexity question in justification logic. In *Proceedings of the 18th international conference on Logic, language, information and computation*, WoLLIC'11, pages 8–19, Berlin, Heidelberg, 2011. Springer-Verlag.
2. Antonis Achilleos. Modal logics with hard diamond-free fragments. *CoRR*, abs/1401.5846, 2014.
3. Antonis Achilleos. On the complexity of two-agent justification logic. Technical Report TR-2014003, CUNY Ph.D. Program in Computer Science, January 2014.
4. Sergei Artemov. Operational modal logic. Technical Report MSI 95–29, Cornell University, December 1995.
5. Sergei Artemov. The logic of justification. *The Review of Symbolic Logic*, 1(4):477–513, December 2008.
6. Sergei Artemov and Roman Kuznets. Logical omniscience as a computational complexity problem. In Aviad Heifetz, editor, *TARK*, pages 14–23, 2009.
7. Melvin Fitting. The logic of proofs, semantically. *Annals of Pure and Applied Logic*, 132(1):1–25, February 2005.
8. Nikolai V. Krupski. On the complexity of the reflected logic of proofs. *Theoretical Computer Science*, 357(1–3):136–142, July 2006.
9. Roman Kuznets. On the complexity of explicit modal logics. In Peter Clote and Helmut Schwichtenberg, editors, *CSL*, volume 1862 of *Lecture Notes in Computer Science*, pages 371–383. Springer, 2000. Errata concerning the explicit counterparts of \mathcal{D} and $\mathcal{D}4$ are published as [11].
10. Roman Kuznets. *Complexity Issues in Justification Logic*. PhD thesis, CUNY Graduate Center, May 2008.
11. Roman Kuznets. Complexity through tableaux in justification logic. In *Plenary Talks, Tutorials, Special Sessions, Contributed Talks of Logic Colloquium (LC 2008), Bern, Switzerland*, pages 38–39, 2008.
12. Roman Kuznets. Self-referentiality of justified knowledge. In Edward A. Hirsch, Alexander A. Razborov, Alexei L. Semenov, and Anatol Slissenko, editors, *CSR*, volume 5010 of *Lecture Notes in Computer Science*, pages 228–239. Springer, 2008.
13. Alexey Mkrtychev. Models for the logic of proofs. In Sergei Adian and Anil Nerode, editors, *Logical Foundations of Computer Science, 4th International Symposium, LFCS'97, Yaroslavl, Russia, July 6–12, 1997, Proceedings*, volume 1234 of *Lecture Notes in Computer Science*, pages 266–275. Springer, 1997.
14. Eric Pacuit. A note on some explicit modal logics. In *Proceedings of the 5th Panhellenic Logic Symposium*, pages 117–125, Athens, Greece, July 25–28, 2005. University of Athens.
15. Tatiana Yavorskaya (Sidon). Interacting explicit evidence systems. *Theory Comput. Syst.*, 43(2):272–293, 2008.